Burroughs Corporation

INTER-OFFICE CORRESPONDENCE

TO:	NAME: Distribution	DATE: September 19, 1984
	CORPORATE UNIT:	DEPARTMENT & LOCATION:
FROM:	NAME: Duncan McClain	PHONE: BENTEL: 2-9229
	CORPORATE UNIT: WW Mktg.	DEPARTMENT & LOCATION: Marketing & Planning-Medium Systems/2A31 - WHQ

SUBJECT: **B2/3/4000 DMSII OPERATIONAL CONSIDERATIONS** cc: LOCATION:

L. Bell (memo only)
S. P. Baxter (memo only)

Distribution

T. DeReus	D. Maddox	M. Prusha	J. Bamberger
B. Fulton	W. Lapinsky	R. Radice	J. Kramer
Garren	P. Link	E. Redfield	

The attached B2/3/4000 DMSII TIP is excellent information that assists greatly in the motivating of B2/3/4000 system users to step up to the benefits of DMSII. Please ensure its distribution to your Marketing Support and Software Services staffs.

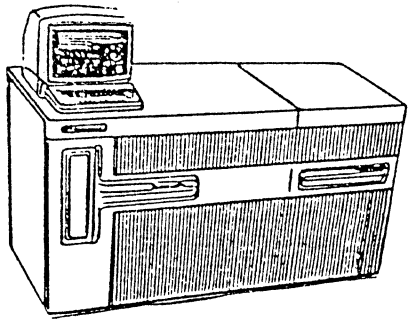
Respectfully,

Duncan McClain
Sr. Marketing Specialist
Worldwide Marketing

DM/smh

Attachment

RECEIVED
SEP 24 1984
COMPILERS &
DATA MANAGEMENT
6700



Customer Support Center South

Burroughs

Technical Information Paper

DATE: AUGUST 17, 1984

PRODUCT: B2000/B3000/B4000 DMSII

SUBJECT: OPERATIONAL CONSIDERATIONS FOR DMSII DATABASES ON B2000/B3000/B4000 SYSTEMS

ORIGIN: CSC -SOUTH
2900 WOODCOCK BLVD.,
SUITE 100
ATLANTA, GA 30341

GA WATS: 1-800-282-1973
US WATS: 1-800-241-7583
BENTEL: 455-0111
TWX: 810-757-0219
DEX: 404-457-1935

AUTHOR: CHUCK LANHAM

This paper will outline general operational requirements and procedures for maintaining a DMSII database on the B2000/B3000/B4000 systems. The emphasis will be on general backup and recovery procedures and will develop a methodology to assist the user in database management.

TO: BMG Communication Code Doc 12, Doc 13, Doc 14

FOR INTERNAL USE ONLY

Printed in U.S.A.

1175288-202

BACKGROUND

A tremendous volume of information has been written to assist the Database Administrator (DBA) in the design, development, and details of a database. Much of the information available has been directed towards the user who has highly trained database personnel on staff, who design and create the database from scratch, and who has a desire and need to know the intricacies of database design.

As database management has evolved, many new tools to assist the DBA in designing and creating the database have evolved as well. Database generators, such as LINC, have relieved the user of having large staffs of database analysts. The user is no longer required to have any knowledge of database design techniques or of the database language, DASDL, in order to generate a database.

Additionally, users are purchasing generated database packages from sources outside their shop. These packages are often pre-compiled and developed outside the users environment. Once installed, tested, and put into the production environment, the user is responsible for the integrity of the database.

The new database users have frequently little or no formal knowledge or training in a database language (DASDL) and the associated operational details of the database. Hardware and software failures do occur, unfortunately, and when the database must be recovered from such an occurrence the new user may not be able to successfully accomplish recovery. This may occur due to lack of proper backup procedures or operational errors in the recovery process.

Another aspect of the higher level generation of the database is the inability of the user to make certain fundamental changes to the physical environment of the database. This may be due to a limitation in the database generator or merely a lack of specific knowledge on how to accomplish this by the user.

With this background, it becomes apparent that there are several specific areas where the use of a database generator or database package may not provide the user with the necessary tools and/or knowledge to maintain the database. This paper is an attempt to provide the user with the necessary guidelines to properly handle the backup and recovery of the database.

INTRODUCTION

This paper is divided into several major topics that should be of interest to any DMSII user. The first section describes a short list of manuals that should be considered required publications for any database user. The next section briefly describes the DMSII environment, the various files required, their purpose, and how they should be maintained. The third section covers backup procedures and provides a rationale and some direction toward establishing sound backup policies. A short description of the two methods for changing the database, reorganization and update, follow. Another major topic, recovery, is covered in the following section. The various types of recovery are explained as well as the purpose and times to use each type. The following section considers a number of MCP options that the user should implement prior to using a DMSII database. A section on the necessary documentation for problem resolution follows. Finally, a short bibliography of database related documentation is given that contains additional information on some of the topics discussed in this paper.

DOCUMENTATION

Although an indepth knowledge of DMSII and DASDL constructs and syntax is not required for LINC or generated database users, the user should have the following documents on hand for quick reference:

1108925	B2000/B3000/B4000 Data Management System II User's Manual, August 1982
1108925-001	PCN to User's Manual for 6.7 release.
ASRDOC 6.7	B2000/B3000/B4000 Systems Software Release Letter (ASRDOC) for the release currently operating on the system.
1138625	B2000/B3000/B4000 Systems Software Operations Guide Volume 1 for the release currently operating on the system.
1138633	B2000/B3000/B4000 Systems Software Operations Guide Volume 2 for the release currently operating on the system.
1090735	B2000/B3000/B4000 Series COBOL ANSI-74 Reference Manual
1090735-001	PCN to COBOL Reference Manual
1090735-002	PCN to COBOL Reference Manual
1090768	B2000/B3000/B4000 Series REPORT PROGRAM GENERATOR (RPG) Reference Manual
1090768-001	PCN to RPG Reference Manual
1090768-002	PCN to RPG Reference Manual
1090768-003	PCN to RPG Reference Manual

This user should be familiar with the sections that deal with database recovery and utility operations, which for the most part can be found in the appendices of the DMSII Users Manual. Updated information regarding the current release can be found in the Systems Software Release Letter (ASRDOC). This list should not be construed, by any means, to be an exhaustive one on the subject of database operations. Many other publications can be found that will assist the user in successfully operating in a database environment. Some suggested additions can be found in the bibliography at the end of this paper.

DMSII SYSTEM

A database is created by the DASDL compiler. This is true even if the user is generating the database thru LINC or other database system generators. The DASDL compiler utilizes a great many files in generating the database. It also creates quite a few in the compilation process. The user should be well aware of the use and maintenance requirements of all these files.

DASDL FILES

The 6.7 DMSII system tape contains 73 files. Because the number of files vary from release to release, the user should consult the ASRDOC on the MCP release tape for the number and names of new or changed files. In general, the DMSII system files can be separated into three groups: (1) Compiler, (2) ICMs, and (3) the Utilities. The compiler group consists of DASDL, DASDLC, and with 6.7, DASDL9. The ICMs are independently compiled modules that the DASDL compiler binds together to form a single program called the database program (DBP). They are identified by the naming convention DM===n where n is the release number. For example, the files identified as DM===7 are the independently compiled modules (ICMs) for the 6.7 release. For release 6.6 these files were identified by DM===6. The utility programs vary from release to release. These utilities make up the remainder of the system. They are extremely important to the user and all users should be made aware of their functions and how to use them.

The compiler, DASDLC, and ICM files must be present for the compilation and creation of the database, or whenever the database needs to be reorganized or updated. Once the database has been successfully compiled and created, the compiler, DASDLC, and ICM files are no longer needed for database operations. If the user has a limited amount of disk space, these files may be removed for normal operations.

On the 6.7 release there are two DASDL compilers, DASDL and DASDL9. They are operationally the same compiler. The only differences are in the amount of memory required and speed of compilation. DASDL9 is a flat (fewer overlays) version of DASDL. For this reason it requires 500KD of memory vs. 200KD for DASDL. LINC 10.3 utilizes the DASDL9 compiler. Only the compiler used need be present on the system during compilation.

Since the ICMs have been previously compiled, a DC will return the compile date of each ICM. It is extremely important that the correct ICMs be present whenever the database is compiled. Since these ICMs are independently compiled, object patch fixes are seldom available. Rather, whenever a problem occurs, new ICMs will be made available through distribution to correct the problem. These new ICMs will have a later compile date and should always be used in place of the original ICMs.

The utilities are well documented in the DMSII User's Manual and their function and operation are beyond the scope of this paper. The user should become familiar with the purpose and methods of execution as soon as possible.

DATABASE FILES

DASDL creates a variable amount of files during the compilation phase of the database. A general flow of the compilation phases is described in the DMSII User's Manual, Section 1. It is extremely important that the user be familiar with the files created in order to properly backup the database for recovery, update, and reorganization. The following structures (files) are created by the DASDL compiler and together make up the database:

xxxDDF - Data Definition File, used by the DASDL and COBOL compilers. It contains a complete description of every structure in the database. It is also needed whenever the database is reorganized and is used by the DMSII utilities as well as the DMS INQUIRY program BLDINQ. It resides, by default, on system resource pack. If it is desired to have the xxxDDF file on a specific pack or on disk it must be file equated at DASDL compile time to that pack. The syntax for accomplishing this is described in the DMSII User's Manual, Section 9. All COBOL program sources must have the xxxDDF file equated as well.

xxxCNF - Control File, contains current information about every structure in the data base. This information is compared against the first block in each database structure to insure that all the structures are the correct versions. The location of the control file is dependent upon the family name of the database identifier (DBP). The default is system resource pack.

xxxMEM - Memory File, (new for 6.7 release) contains various tables used by the Database Program (DBP). This file is read in at BOJ as part of the database program initialization process. It resides on the same pack as the xxxCNF file.

xxx001 - Statistic File, this file will only be present if the STATISTICS option was set in the DASDL source. This file resides on the same pack as the xxxCNF file.

xxxnnn - Structure Files, these are the database files. The files will be named xxx003 thru xxxnnn. The xxx002 structure is not currently created as of the 6.7 release. These structure will reside on either disk, system resource pack, or a specific family as declared in the DASDL source. The default is system resource pack.

xxxICM - Database Program (DBP) template, used during the bind phase which creates the DBP. It resides on disk. During this phase a printer backup listing is created that contains addresses for the various ICMs bound into the DBP. This listing MUST be saved and backed up along with the database. The listing name should be changed to xxxBND.

The [xxx] are the first three characters of the Database Program. For example, if the DBP is named PERDBP then all the files will begin with PER (e.g. PERICM, PERDDF, PERCNF, PER001, etc.).

```
**************************************************************
*    IMPORTANT TIP:    Always  BACKUP the generated BIND LIST  *
*    after every DASDL compile.  On 6.7 change printer backup  *
*    number to xxxBND and it will be backed up  automatically  *
*    during any offline or online dump.  For example:          *
*                                                              *
*                      CH @00010 PERBND                        *
*                    or CH @00010 ON PACK                      *
**************************************************************
```

xxxMEM - Memory File, (new for 6.7 release) contains various tables used by the Database Program (DBP). This file is read in at BOJ as part of the database program initialization process. It resides on the same pack as the xxxCNF file.

xxx001 - Statistic File, this file will only be present if the STATISTICS option was set in the DASDL source. This file resides on the same pack as the xxxCNF file.

xxxnnn - Structure Files, these are the database files. The files will be named xxx003 thru xxxnnn. The xxx002 structure is not currently created as of the 6.7 release. These structure will reside on either disk, system resource pack, or a specific family as declared in the DASDL source. The default is system resource pack.

xxxICM - Database Program (DBP) template, used during the bind phase which creates the DBP. It resides on disk. During this phase a printer backup listing is created that contains addresses for the various ICMs bound into the DBP. This listing MUST be saved and backed up along with the database. The listing name should be changed to xxxBND.

The [xxx] are the first three characters of the Database Program. For example, if the DBP is named PERDBP then all the files will begin with PER (e.g. PERICM, PERDDF, PERCNF, PER001, etc.).

```
**********************************************************************
*    IMPORTANT TIP:    Always  BACKUP the generated BIND LIST        *
*    after every DASDL compile.  On 6.7 change printer backup        *
*    number to xxxBND and it will be backed up  automatically        *
*    during any offline or online dump.  For example:                *
*                                                                    *
*                        CH @00010 PERBND                            *
*                     or CH @00010 ON PACK                           *
**********************************************************************
```

BACKUP PROCEDURES

This is one of the most critical areas of database management and often the area where problems develop. Proper backup of the database is essential for recovery or for making any changes to the database. **If backup procedures aren't performed in a proper manner, the user may not be able to successfully recover or make future changes to the database.**

The database must be backed up to reliable media after any change is made to the database via reorganization or update. Every structure in the database contains time and date information which will change when changes are made to that structure. This information is also reflected in the xxxCNF file and is verified whenever the database is opened. If this time and date information does not match, the database will abort with a time and date mismatch. Therefore, it is recommended that **BEFORE** and **AFTER** each reorganization or update to the database, a complete backup be performed. Since the database usually contains critical information it is imperative that if disaster strikes, the user can back up and start again without losing any valuable data.

Additionally, **the database should be backed up on a DAILY basis.** Prior to the 6.7 release, this could have presented some scheduling conflicts since only an offline dump was available through the use of a utility such as SYSTEM/COPY or LOADMP. With the 6.7 release, both offline and online dumps of the database can be performed through the IR command on the ODT. The procedures for performing both types of dump are outlined in the DMSII User's Manual and the ASRDOC for the 6.7 release. The user should read this information and become thoroughly familiar with the syntax and specific requirements of either the offline or online dump.

It is highly recommended that all operations personnel be thoroughly familiar with all the backup procedures. The extreme importance in the integrity of the database itself demands perfection. Reliability is essential and complete verification of the backup should always be performed. Detailed written instructions and procedures must be implemented and strict supervisory control placed on all backup policies. One small error can result in many hours lost and significant costs incurred in an attempt to recover from a disaster.

An additional feature of the 6.7 Offline/Online dump is that the dump contains the location (DISK, PACK, [family name]) where each structure or file physically resides. When a load is performed, these files are placed on the correct media, automatically. Prior releases required several separate LOAD or COPY executions to place files on different media. This required a more complex set of operational instructions, leading to errors in backup and loading procedures.

Both the offline and online dumps will backup the following database files:

1. **Database program (DBP)**
2. **xxxICM**
3. **xxxCNF**
4. **xxxMEM**
5. **xxx001**
6. **xxxDDF**
7. **xxx003 thru xxxnnn structure files**
8. **xxxBND bind listing if present**

It is important to note **that no audit files are dumped** by either of these dumps. **The user must backup the audit trail by using a utility such as SYSTEM/COPY.**

The audit trail consists of audit files that follow the naming convention: xxxAnn thru xxxMnn and xxxNnn thru xxxZnn. The audit trail begins with xxxA01 and increments thru xxxA99. The next file created will begin with xxxB01 and increment thru xxxB99, etc. When xxxM99 is full, the next one created will be xxxA01 again. If duplicated audit is used then the duplicate audit will be the series xxxNnn thru xxxZnn.

The audit trail should be backed up daily as well. It is imperative that the audit trail be verified, using DMSAUD, prior to backing it up. Once the audit trail has been verified and backed up, all the audit files should be removed. If not removed, the last audit file used by the database program will be reopened and updated by the DBP at the next BOJ. If the audit trail is later reloaded from the backup, the older version will be loaded over the more current version. The audit trail will no longer be complete and subsequent recovery attempts will fail.

At a minimum, the user should:

1. **Verify the audit trail daily using DMSAUD.**
2. **Back up the audit trail daily and verify the backup.**
3. **Maintain a verified copy of the complete audit trail (offsite if possible).**

Backing up the database with less than perfect procedures is truly an exercise in futility. Recovery time is not the time to find out that the backup was performed incorrectly or not at all. If the data contained in the database is important, then the backup should be considered as equally important. The audit trail should be verified prior to backup using the DMSAUD utility. Insure that all personnel involved in the recovery process are familiar and have actually tested and practiced all recovery and backup procedures before they are required.

For more detailed information concerning the capabilities and syntax of ONLINE/OFFLINE DUMPING, consult the 6.7 ASRDOC, Section 12 or the DMSII Users Manual, Appendix B.

```
**************************************************************
*  A GENTLE REMINDER:    Always  make sure  that the backup  *
*  contains "every" file in the database.  Version mismatch  *
*  errors will definitely ruin one's day.                    *
**************************************************************
```

REORGANIZATIONS AND UPDATES

Changes can be made to the database by either of two methods, REORGANIZATION or UPDATE. Reorganization is used when physical changes to the database files and records are needed. Update is used to make changes to the database which do not affect the physical format of the files or records. LINC always uses REORGANIZATION. Reorganization produces 2 database ICMs (xxxICM and xxxRCM) and 2 database programs (xxxxxx and xxxREO). Once the DASDL compilation is finished, the database is marked as "reorganization required" and cannot be accessed until the reorganization has been completed. This is accomplished by the IR command, or if the user is on 6.7, through DMSUTL and the DMSDLR option. LINC 10.3 release will accomplish this through the $SET AUTOREORG compiler control card.

Once this process is completed, and prior to any program opening the database the database **must be backed up.** This will be the new starting point for any future database recovery.

LINC 10.3 requires the use of the 6.7 releases of MCP and DMSII in order to run. An existing database (6.6) must be upgraded to 6.7 by using the $UPDATE DASDL compiler option. $REORGANIZE cannot and must not be used to upgrade from 6.6 to 6.7 release. For the LINC user, the 10.3 release letter gives detailed instructions on the use and execution of LNCUPD, which will accomplish the update. DO NOT attempt to REORGANIZE an existing 6.6 database using the 6.7 DMSII release.

****
* **REMEMBER!** Any time the database is modified, either by an *
* update or a reorganization, the prudent user will back up *
* the complete database, both BEFORE and AFTER performing the *
* reorganization or update. The time spent in safe-guarding *
* the organizations database from disaster will prevent time *
* wasted attempting to recapture lost data. There are only *
* three things to consider when a change is to be made to *
* the database: *
* *
* **BACKUP! BACKUP! BACKUP!** *
****

RECOVERY

Recovery is another critical aspect of database operations. When mistakes are made or errors occur, the database must be brought back to a valid point in time. There are currently five types of recovery. The type required and how it is implemented will be dependent on the specific circumstances involved, but in general can be described as follows:

ABORT RECOVERY..if a program which is updating the database fails abnormally while in transaction state, the system will attempt an ABORT recovery automatically. No operator intervention is required.

HALTLOAD RECOVERY..occurs after a system crash or DBP DS/DP. This will be automatically initiated at the next BOJ of the database program or can be initiated by the operator through the ODT by the IR command.

REBUILD RECOVERY..allows the user to move the database forward from a previous dump of the database through some specified stopping point. This type of recovery is initiated by loading out a previously backed up database and specifying REBUILD in IR command on the ODT. The user specifies the stopping point. The default is the end of the audit trail. Since there is no way of determining the last file in the audit trail, the user must use the OF keyboard command when the DBP hangs on a NO FILE condition. This will terminate the REBUILD.

RECONSTRUCT RECOVERY..allows a specific structure or structures to be reconstructed from an old structure file. Reconstruct is used to recover damaged database files. Only a portion of the database is recovered. Prior to initiating a reconstruct recovery, backup copies of the damaged file(s) must be loaded as well as a copy of the control file (xxxCNF) as it existed when the backup was taken. This file must then be named xxxCNO. The current xxxCNF file must also be present. The starting and stopping points are determined by the old (xxxCNO) and current (xxxCNF) control files.

MCP CONSIDERATIONS

When bringing up a DMSII database for the first time, several MCP options must be specified. The DMS2 extension module must be set by either the SO DMS2 keyboard command or the USE card in the Cold/Warm Start deck. Two LIMIT cards must be inserted into the Cold/Warm Start deck. These are:

LIMIT DMSDBP [integer]

This card specifies the maximum number of DBPs allowed on the system at one time. The [integer] must be between 1-9. Always specify a number greater than will normally be run to allow for test databases, etc. Since this table requires only a small amount of memory, the cost of setting this to the maximum of 9 should not be excessive and will eliminate the necessity of PERFORMING a coldstart or warmstart to add an additional DBP to the system.

LIMIT DMSUSERS [integer]

This card specifies the maximum number of programs allowed to access the database at one time. When determining what this should be, always plan ahead for expansion. The DBP itself will use one entry in this table as well. The default for this LIMIT is 0, therefore it must be specified in the coldstart/warmstart deck.

If neither of these LIMIT cards are specified, a ODT message will display DMS TABLE OVERFLOW and the database will not be allowed to run.

****
* **NOTE OF CAUTION:** If the system DATE/TIME is changed, make *
* sure that it is changed to a more recent DATE and/or TIME. *
* Otherwise, if the database were updated after the new date *
* and time, complete recovery may not be possible. For *
* example, assume an original date of 060284, which should *
* have been 060184. If the database were updated while using*
* the incorrect date, changing the date back will cause a *
* recovery attempt using this audit trail to be unsuccessful.*
* *
****

DATABASE ERROR REPORTING PROCEDURES

When a problem occurs in either the compiler or during the execution of the database, the user should contact his local Burroughs Representative, or the Customer Support Center. The information that will be necessary to assist in the resolution of the problem varies but in general, the following should be considered the minimum.

COMPILER PROBLEMS

If the DASDL compiler fails, it will usually take the form of either a FATAL ERROR condition, or a compiler failure which causes the compiler to terminate with a memory dump.

If the compiler quits with a FATAL ERROR, a message will be output in the compile listing. If this occurs and the problem cannot be resolved, the following information and documentation will be required:

a) The DASDL compile listing.

b) Machine readable source necessary to duplicate the problem. If possible, a small test case is preferred.

c) The xxxDDF file.

d) The xxxICM file.

e) A SPOLOG containing the failure message.

If the compiler terminates abnormally with a compiler failure, then the following information will be required:

a) A Memory dump of the DASDL compiler containing the failure.

b) The DASDL compile listing (if one is produced)

c) The DASDL source submitted to the compiler.

DMS EXECUTION-TIME PROBLEMS

It should be standard procedure that whenever a DBP failure occurs, a backup dump of the database should be immediately performed. This may seem unduly time-consuming, but more often than not, the problem will be resolved much faster if the current state of the database is preserved for analysis. If a problem arises during the execution of the DBP (undigit arithmetic, address errors, etc.), which cause the DBP to terminate, the following information and documentation will be required:

a) An interpreted DBP Dump. This will be automatically produced when "DBPDMP" is zipped by the Dump Analyzer Program (PM [DBP dump #]). The interpreted dump can also be obtained by execution of the DBPDMP program.

b) The current bind listing (xxxBND). It is **absolutely essential** that this listing be available. This bind list is basically a map of the bound DBP object code file and is a necessary tool for debugging purposes. Whenever the DBP is compiled, this listing must be backed up with the database and preserved until the database program is changed again. Without the bind list the problem cannot be resolved. This is produced by DASDL during the bind phase of the compile and the backup procedure has been previously described. A printed copy should be readily available for the most expeditious resolution to the problem.

c) The current DASDL source and compile listing. The source must be "machine readable" for recompilation.

d) SPO log of the failure time frame. This provides accurate information about the error.

e) If the DBP is "hung", i.e., programs are waiting DMS function and the DBP is also waiting, then a MCP dump (0 DM ALL), prior to dumping the DBP will be required.

f) Certain database problems may require that the entire database be submitted.

g) Transaction failures (resulting in DBP failures) must include all of the above, as well as the program that was used to initiate the transaction. Additionally, a copy of the audit files on tape may be required.

This information should be available whenever requesting assistance. Although the problem may many times not require all of the above items, the amount of time taken to properly document the problem most often will result in a more expeditious resolution to the problem. If a FCF is required to be submitted, then it can be done without delay. For any of the machine readable source or backup of the database files submitted, the media sent must be 1600 BPI PE tape format. To insure return of the media, attach complete name and address to the media submitted.

Further information on problem reporting can be found in the following publications:

1130838	User Guide to Program Product Fault Reporting
1137965	B2000/B3000/B4000 Product Support Information Manual
ASRDOC 6.7	System Software Release Letter ASRDOC 6.7 Section 12

****
* **A FINAL WORD:** Whenever the database is reorganized, *
* there will be three printer backup listings produced. *
* The first will be the DASDL source listing. The next *
* one will be the bind listing for the reorganize DBP. *
* The third listing will be the bind listing for the *
* regular DBP. Both the bind listings should be backed *
* up. Change the second listing to xxxBNR and the last *
* listing to xxxBND as previously described. Save the *
* xxxBNR listing until the next reorganization. *
* *
****

BIBLIOGRAPHY

1) Trout, F., and Meyer, D.; "Implementation Considerations"; BURROUGHS Technical Information Paper # 1114238-016, May 28, 1981

2) Trout, F.; "Design Primer For On-line Database Systems"; BURROUGHS Technical Information Paper # 1114238-019, July 25, 1983

- - - - - -TABLE OF CONTENTS- - - - - -

PAGE

BACKGROUND 2

INTRODUCTION 3

DOCUMENTATION. 4

DMSII SYSTEMS. 6

  DASDL FILES 6

  DATABASE FILES 7

BACKUP PROCEDURES 9

REORGANIZATIONS AND UPDATES 12

RECOVERY 13

MCP CONSIDERATIONS 15

DATABASE ERROR REPORTING PROCEDURES 16

  COMPILER PROBLEMS 16

  DMS EXECUTION-TIME PROBLEMS 17

BIBLIOGRAPHY 19